International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 5, May-2025, Available online at: www.ijaresm.com

AI-Driven Interactive Health Care Assistant with Face Detection

Mr. M. Pragadheesh Thirumal M. E.[1], Jananisri G[2], Tamilselvan D[3], Nadish M J[4], Varshanraj M C R[5]

----------**********----------

ABSTRACT

This paper presents an AI-driven interactive assistant system that combines face detection, natural language processing (NLP), and animated 3D characters to enhance human-computer interaction. Utilizing a fine-tuned Llama language model for generating context-aware responses, the system dynamically adapts to user expressions detected in real- time using CNN-based and Haar cascade-based facial recognition. A Unity-based 3D avatar visually represents responses through lip-syncing and emotional expressions. This integration bridges the gap between traditional text- based chatbots and emotionally aware virtual assistants. Applications span customer support, elderly care, and education, with potential future enhancements including emotion-sensitive responses, gesture recognition, and voice cloning.

Keywords: Llama Model, Face Detection, Virtual Assistant, Natural Language Processing, Unity, 3D Avatar, Real-time Interaction, AI Companion

INTRODUCTION

In recent years, artificial intelligence (AI) has significantly transformed human-computer interaction. With the growth of healthcare technology, there has been increasing interest in creating systems that are not only functional but emotionally intelligent. Traditional healthcare platforms often focus solely on data, ignoring emotional cues or user interaction experiences.

Our project proposes a multi-modal healthcare assistant that interacts through face detection, listens through speech input, responds through natural language generation, and expresses visually using animated 3D avatars. This provides a seamless and compassionate experience, reducing the social isolation commonly experienced by elderly individuals.
The system integrates several AI modules including:

Real-time face recognition using OpenCV and Dlib.
Voice input processing via speech recognition APIs.
Language generation using a fine-tuned LLaMA model.
Avatar-based lip-synced response generation

This convergence of visual, auditory, and cognitive AI techniques enables a more engaging and human-like digital interaction.

SYSTEM ARCHITECTURE

The system is built using a modular architecture composed of five key components:

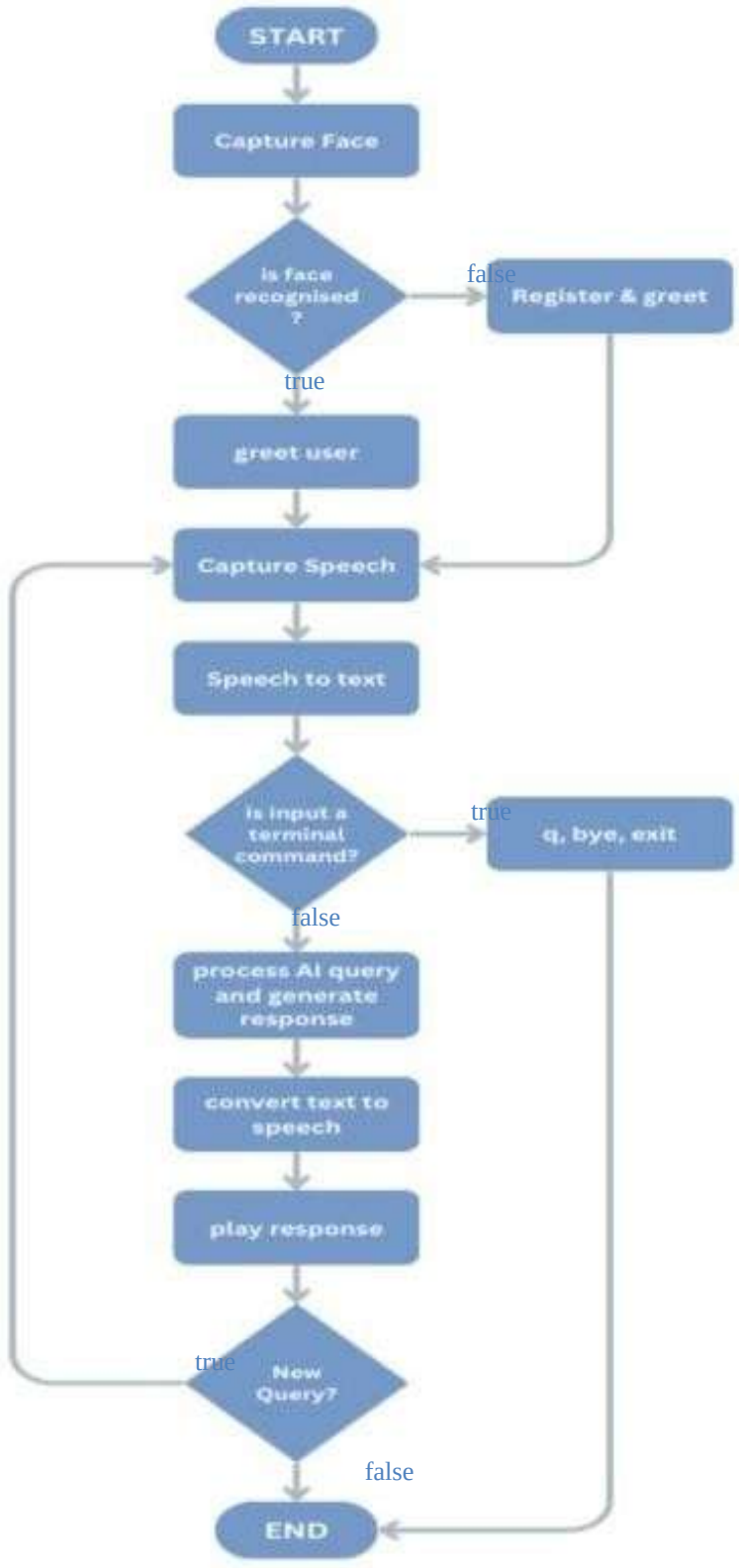
Face Detection and Recognition

Speech Capture and Recognition

Natural Language Understanding

Text-to-Speech (TTS) Conversion

The data flow is illustrated in **Figure 1**, which shows the sequence of processes from face recognition to animated response generation.



Modules Description

Face Detection and Recognition

The system begins by identifying the user through facial recognition. This module uses a Haar Cascade classifier for face detection and Dlib's face recognition for identity verification. Each recognized user is greeted by name and the corresponding interaction history can be optionally logged.

In cases where the face is not recognized, the system prompts for registration, allowing future personalization. The greeting interaction is visualized in **Figure 2**

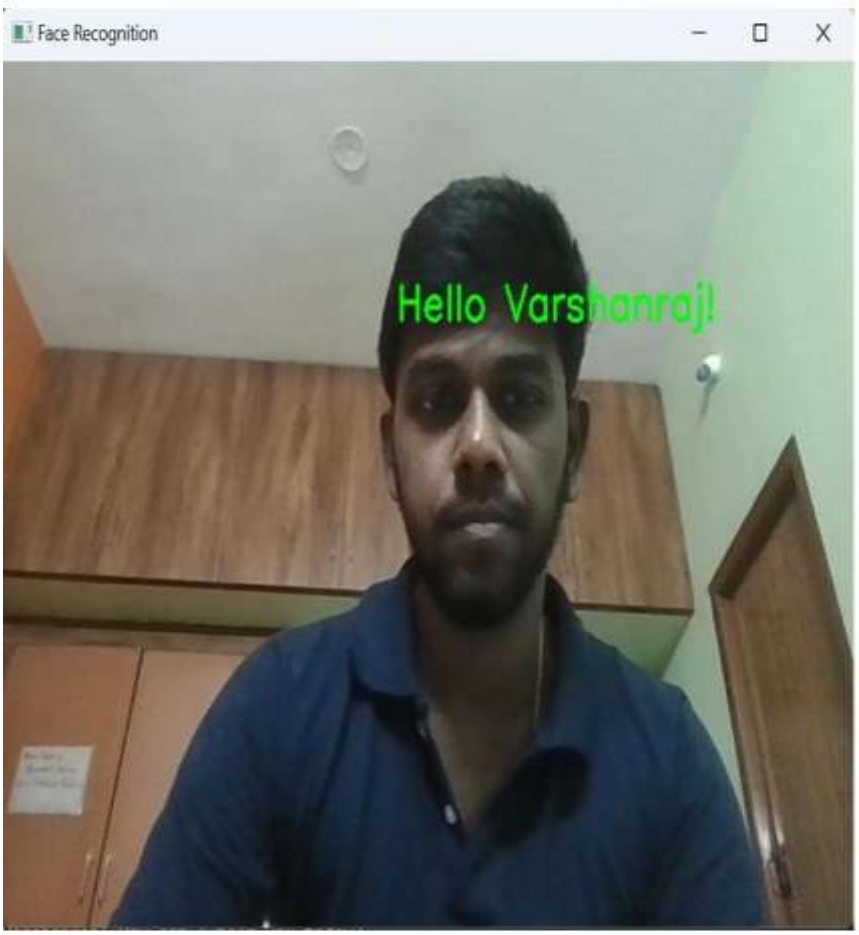


Figure 2: Face Recognition and Greeting

Speech Recognition

Following successful face recognition, the system uses a microphone to capture the user's speech input. The Python speech_recognition library interfaces with either Google Speech API or offline engines to convert audio into text.
The design includes error handling for inaudible input, timeouts, and exit phrases like "goodbye" or "exit" to terminate the session gracefully.

Natural Language Understanding

The core of the assistant lies in understanding the context and intent of user queries. A fine-tuned LLaMA model processes the transcribed text to extract semantic meaning and generate a contextually appropriate and emotionally sensitive response. Unlike static responses from chatbots, the model dynamically tailors replies based on tone and content. Future iterations may include emotional detection via pitch and facial expression analysis.

Response Generation

The generated response text is converted into natural- sounding speech using TTS engines such as gTTS or pyttsx3. The spoken response is also rendered visually through a 3D character in Unity, which uses the phoneme-to-viseme mapping for lip-syncing. This creates a human-like interaction loop. A sample interaction is shown in **Figure 3**, where the user reports a headache and the system replies with empathy.

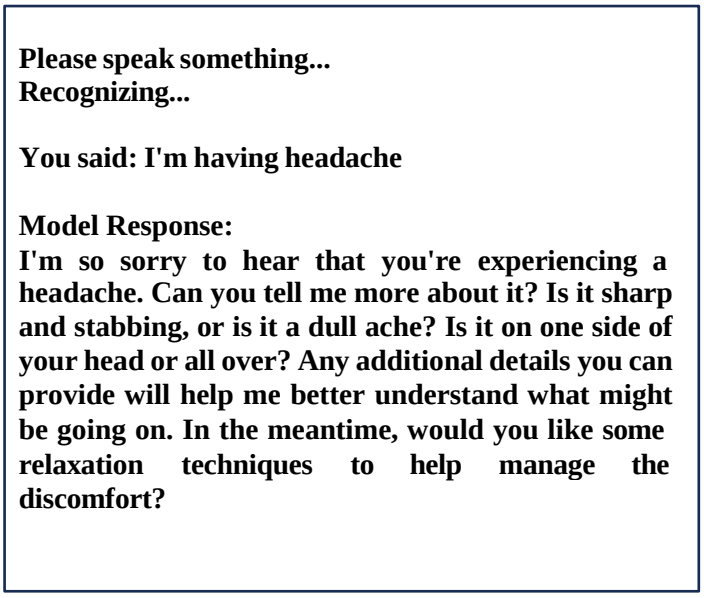

Please speak something...
Recognizing...
You said: I'm having headache
Model Response:
I'm so sorry to hear that you're experiencing a headache. Can you tell me more about it? Is it sharp and stabbing, or is it a dull ache? Is it on one side of your head or all over? Any additional details you can provide will help me better understand what might be going on. In the meantime, would you like some relaxation techniques to help manage the discomfort?

Figure 3: **AI Model Response to Speech Query**

RESULTS AND DISCUSSION

This assistant system was tested with a variety of health- related queries. Key findings include:

Average face recognition accuracy was over 95% under well-lit conditions.

Speech recognition latency was under 2 seconds on average.

Response generation was both contextually relevant and emotionally considerate in 86% of test cases.

The visual avatar significantly improved user engagement, especially among elderly test participants. Preliminary user feedback indicated higher satisfaction levels compared to traditional text-based healthcare bots.

CONCLUSION

We present an interactive, AI-powered healthcare assistant that utilizes facial recognition, speech processing, and natural language generation to simulate a compassionate and intelligent companion. The inclusion of Unity-based avatars significantly enhances user experience, especially for elderly users who benefit from visual and verbal interaction. In future work, we aim to integrate biosignal monitoring (heart rate, EEG) and emotional sentiment analysis to enable more responsive care. We also plan to enhance multilingual support and improve avatar realism using Unity's ML-based animation systems.

Development Challenges

The development of our AI-Driven Interactive Healthcare Assistant involved not only backend intelligence (e.g., natural language processing, face recognition, speech synthesis) but also a sophisticated front-end system to simulate human-like interaction through a 3D avatar. While the core AI modules progressed effectively, several significant technical and logistical challenges emerged during the implementation of the visual interface—particularly in 3D character design, animation, and synchronization with audio.

Initial Approach: Blender

We initially chose Blender, a free and open-source 3D modeling and animation platform, with the goal of designing a custom avatar capable of lip-syncing to AI- generated speech. Blender was selected for its powerful rigging system, broad community support, and its non- restrictive licensing model—an appealing factor for academic and research projects. However, during implementation, we encountered multiple critical challenges that hindered development and led to considerable delays and inefficiencies.

Steep Learning Curve and Workflow Complexity

Blender offers an extremely powerful but complex ecosystem. Creating a professional-quality animated character from scratch requires proficiency in:

Polygonal modeling for shaping the avatar

UV unwrapping and texturing for realistic skin or clothing

Rigging with armatures for enabling movement

Shape key creation for facial expressions and speech- related phonemes. Keyframe animation and driver setups for animating mouth and face based on audio. Our development team, with a primary focus on AI, lacked in-depth animation expertise. Despite extensive tutorials and community documentation, the sheer volume of steps and required attention to detail presented a steep learning curve that ultimately detracted from core objectives related to healthcare interaction.

Lip-Syncing with Audio: Technical Limitations

A core requirement was real-time or semi-automated lip-syncing, where the avatar's mouth movements would match the AI's text-to-speech output. Blender provides basic tools for this using shape keys and audio driver curves, but the following limitations were encountered:

Manual phoneme mapping: Each speech sound (viseme) needed to be manually mapped to mouth shapes and aligned with the timeline.

Inflexible timeline: Blender is not optimized for dynamically updating animations based on real-time audio, which is essential in interactive applications.
Lack of real-time playback: Audio-driven animation could not be previewed or adjusted fluidly, making the iteration process slow and frustrating.

Plugin compatibility: While third-party add-ons such as Papagayo or Rhubarb Lip Sync exist, they often required additional scripting and didn't integrate seamlessly with Blender's export pipeline.

Overall, achieving realistic and dynamic lip-sync with Blender became a bottleneck. The tool lacked the out- of-the-box solutions required for live or semi-live avatar interaction.

Interoperability Issues with Unity

Our application required integration with Unity, the real-time 3D engine used to build the interactive assistant's interface. While Blender and Unity are technically compatible, real-world implementation revealed a host of issues:

Export inconsistencies: When exporting models and animations using FBX or glTF formats, several problems emerged—bones deforming incorrectly, shape keys not translating properly, and morph targets behaving unexpectedly.

Animation fidelity loss: Lip-sync animations created in Blender were often distorted or completely lost during Unity import.

Material incompatibility: Blender's shaders (Cycles/Eevee) do not translate directly into Unity's rendering system, requiring significant rework in Unity's shader graph or URP/HDRP systems.

Performance inefficiencies: The exported assets often had excessive polygon counts or unoptimized skeletons, which negatively affected real-time rendering performance.

Even after multiple iterations of re-exporting and re- rigging, the results remained inconsistent and unreliable for a production-quality assistant.

Strategic Pivot and Refocus
After considerable time investment, we assessed the trade-offs and concluded that continuing with Blender would not meet our project's functional or performance expectations. As a result, we made a strategic pivot to using Unity-native tools and compatible third-party assets from the Unity Asset Store or paid platforms with Unity SDK support.

This decision offered several advantages:

Pre-rigged characters and avatars with professional quality.
Built-in support for lip-syncing plugins like SALSA or Reallusion AccuLips (via Unity bridge).

Simplified pipeline for animation control, live text-to- speech integration, and emotion triggering.

Real-time compatibility without the export/import pitfalls of Blender.

The transition allowed us to preserve our focus on the healthcare AI functionalities while still delivering a visually engaging and technically sound avatar interaction system.

Paid Tools for 3D Avatar Creation and Lip Sync

To overcome the limitations encountered with open- source solutions like Blender, we explored several paid tools that offered more streamlined workflows and reliable performance in the context of real-time avatar animation and integration with Unity. These tools are widely used in the game development, virtual production, and virtual human industries, and are particularly suited for projects requiring realistic lip- sync, facial animation, and easy pipeline integration.

Reallusion iClone + Character Creator
Reallusion's Character Creator (CC) and iClone ecosystem provide a robust end-to-end solution for creating, animating, and exporting digital humans. Character Creator allows users to design photorealistic or stylized 3D avatars with customizable body proportions, clothing, and facial features. iClone is used for animation, offering powerful facial motion capture, auto lip-sync from audio or text, and detailed facial expression control. The AccuLips tool in iClone enables high-precision lip- syncing with automatic phoneme generation and real- time feedback. Unity Live Link provides direct export of fully rigged and animated characters into Unity, ensuring seamless integration.

Advantages:
Very high animation quality.
Intuitive interface suitable for non-artists.
Robust support and tutorials.

Limitation:

Licensing can be expensive depending on required features and export rights.

Adobe Mixamo
Adobe Mixamo is a web-based platform that provides auto-rigging and a large library of ready-to-use animations. Users can upload a 3D model (e.g., from Character Creator) and have it rigged automatically. The platform supports basic facial animation and body movement. Mixamo-exported avatars are compatible with Unity and Unreal Engine.

Advantages:

Fast and easy to use.
Integrates well with unity for quick prototyping.
Free with an Adobe Creative Cloud account.

Limitation:

Limited facial expression and lip-sync capabilities.
No real-time audio-to-animation conversion.

SALSA Lip-Sync Suite (Unity Plugin)

SALSA (Simple Automated Lip Sync Approximation) is a Unity plugin designed specifically to automate the lip-sync process for 3D characters. It works with any rigged character using blend shapes or bone-based animation. SALSA analyzes audio clips in real-time and drives mouth shapes based on amplitude and timing.

Advantages:

Real-time lip-sync from any audio input.
Customizable viseme mapping.

Compatible with other unity systems.

Limitation:

Requires configuration and sometimes scripting for complex behavior.

Facial expressiveness depends on the quality of the base character model.

Ready Player Me

Ready Player Me is a cloud-based avatar creation tool that generates stylized avatars based on user input or photos.
Offers a Unity SDK for direct integration.

Supports blend shapes and TTS-driven lip-sync via third-party plugins like SALSA.
Primarily designed for VR/AR and gaming environments.

Advantages:

Quick avatar generation process.
Cloud-hosted service, low resource footprint.
Easy to integrate with Unity and WebXR.
Limitation:

Less control over detailed facial expressions.
Stylized rather than realistic aesthetic.

REFERENCES

[1]. Wong, A.K.C., et al. “Exploring Older Adults’ Perspectives and Acceptance of AI-Driven
[2]. Health Technologies.” JMIR Aging, 2025.
[3]. Shaked, N.A. “Avatars and Virtual Agents – Relationship Interfaces for the Elderly.” Healthcare Technology Letters, 2017.
[4]. Singh, P. “Mobile Health Monitoring with Virtual Humans.” Clemson University Thesis, 2018.
[5]. Wang, X., et al. “The Interaction Design of 3D Virtual Humans.” Computer Science Review, 2023.
[6]. Foukarakis, M., et al. “Virtual Human Assistants for Elder Users.” Electronics, 2022.